

# Treatment of Multibody Interactions in Molecular Simulations of Systems with General Bond Networks

ROBERT E. TUZUN,\* DONALD W. NOID,  
BOBBY G. SUMPTER

*Chemical and Analytical Sciences Division, Oak Ridge National Laboratory,  
Oak Ridge, Tennessee 37831-6197*

*Received 25 September 1996; accepted 11 March 1997*

**ABSTRACT:** One of the most formidable difficulties in the computer programming of molecular simulations is the sometimes complicated bookkeeping required for keeping track of internal coordinates and their derivatives. A completely general method for keeping track of stretch (two-body), bend (three-body), and torsion, wag, and other four-body interactions for ANY bond network is presented. Computer code using this method for calculating internal coordinates and their derivatives can be used for completely different types of bond networks, no matter how complex, with little or no modification. The method is designed to incorporate recent improved formulas for calculating internal coordinates and their derivatives to ensure the most optimal calculation sequence. © 1997 John Wiley & Sons, Inc.\* *J Comput Chem* 18: 1513–1522, 1997

**Keywords:** general bond network; molecular simulation; geometric statement function; internal coordinates; polymer

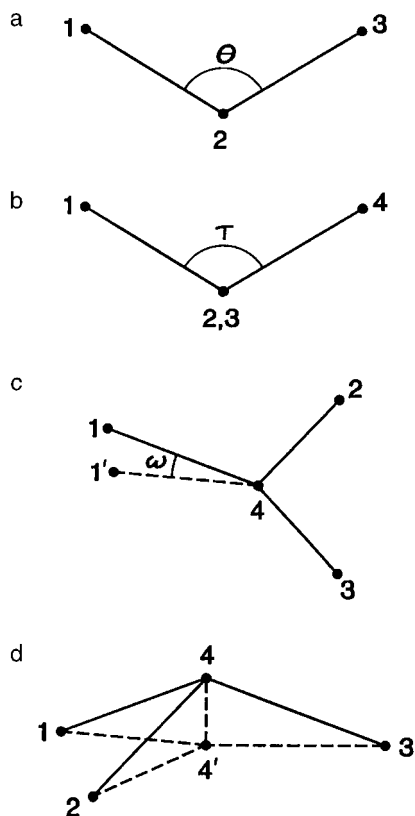
## Introduction

One major difficulty in the computer programming of molecular simulations is the sometimes complicated bookkeeping required to keep track of internal coordinates and their deriva-

tives. The potential energy surfaces of polymers usually contain a large number of interactions between chemically bonded atoms, usually two- (bond stretch), three- (bend), and sometimes four- (torsion, wag, etc.) body interactions (Fig. 1). A fully optimized molecular simulation code is usually specially written for the type of bond network being treated. Even for fairly simple systems such as fully atomistic polyethylene it can be a tedious, error-prone task to program the calculation of stretch, bend, torsion, and nonbonded forces. For

\*Correspondence to: R. E. Tuzun; e-mail: tuzunre@ornl.gov

Contract/grant sponsor: U. S. Department of Energy; contract/grant number: DE-AC05-96OR22464



**FIGURE 1.** Internal coordinates for three- and four-body interactions. (a) Bend angle,  $\theta_{123}$ . (b) Torsion angle,  $\tau_{1234}$ , viewed down the 23 axis. (c) Wag angle,  $\omega_{1234}$ . The point 4' is the projection of point 1 onto the 243 plane. (d) In-plane and out of plane bend angles. The point 4' is the projection of point 4 onto the 123 plane.

example, if each carbon atom is assigned two hydrogen atoms  $H_A$  and  $H_B$ , then four types of CCH bends must be programmed:  $CCH_A$ ,  $CCH_B$ ,  $H_A CC$ , and  $H_B CC$ . Once this code is debugged and optimized it can be difficult and time consuming to adapt it for other uses such as carbon nanotubes.

In most molecular simulations, calculating forces (first derivatives) or at times second derivatives of potential energy terms is by far the most expensive part of the calculation, with derivatives of internal coordinates taking much of this effort. Because of the sheer number of interactions, nonbonded forces often take even more effort to calculate; however, this part of the code can be economized by imposing cutoff distances, by using simple potential energy functional forms (such as Lennard-Jones, which requires no expensive operations such as square roots for forces or energies), or by other

methods.<sup>1</sup> Because of this, in complex bond networks the calculation of forces or second derivatives of bonded interaction potential energy terms still represents an important fraction of the computational effort. Therefore, considerable effort has been spent in optimizing the calculation of derivatives of bonded internal coordinates such as bend and torsion.<sup>2-7</sup>

The geometric statement function (GSF) method by Noid et al.<sup>4</sup> and recent improvements by Tuzun et al.<sup>5</sup> were designed to address the problem of complex bond networks. The essential thrust of the GSF method is to write internal coordinates and their derivatives in terms of other internal coordinates contained therein. For example, a torsion angle sequence 1234 contains two bends, 123 and 234, and three stretches, 12, 23, and 34. A potential energy surface with torsion terms will very likely also contain bend and stretch terms. Use of these quantities and their derivatives, which would be required in other parts of the calculation, renders computation of torsion angles and derivatives efficient and simple to program.

This article describes an efficient method for implementing the GSF method for any type of bond network. Starting from a connection table for a general bond network, this method generates lists of bend, torsion, and other interactions in such a way as to eliminate the duplication of expensive calculations (square roots in bond distances, for example). In addition to ensuring an optimal calculation sequence for internal coordinates and their derivatives, it greatly reduces the difficulty of writing new molecular simulation software or adapting existing software for other uses (branched polymers, carbon nanotubes, and so on).

In the next sections the process for the generation of interaction lists for various internal coordinates is described in detail. This process automates the listing of bend and stretch interactions contained within each internal coordinate, even for the most complex bond networks, as we shall show. Of course, in every molecular simulation the functional form of each potential energy term must still be accounted for; however, the general bond network method can aid in the bookkeeping for this area too.

Considerable savings can be obtained by noting that an internal force acting on particles  $1, \dots, n$  does not move the center of mass of the particles involved (this property is known as translational

invariance). This means

(\frac{\partial V}{\partial q\_1} + \frac{\partial V}{\partial q\_2} + \cdots + \frac{\partial V}{\partial q\_n}) = 0, \tag{1}

where  $V$  is a function of one or more internal coordinates and  $q$  is  $x$ ,  $y$ , or  $z$ . The subscript on  $q$  refers to the particle number. Applying translational invariance and the symmetry condition,

\frac{\partial^2 V}{\partial q\_i \partial q'\_j} = \frac{\partial^2 V}{\partial q'\_j \partial q\_i}, \tag{2}

saves considerable effort for calculating second derivatives.

STRETCH INTERACTIONS

The bookkeeping for stretch interactions in a molecular simulation is simple. Figure 2 shows a bond network chosen for the purpose of illustration and Table I its connection table. Such tables are commonly used, for example, in Brookhaven PDB (Protein Data Bank) files. It is straightforward to generate a list of stretch interactions directly from the connection table and to calculate and store each bond distance  $r$ ,  $1/r$ , and its derivatives. Only three first derivatives per interactions need to be computed because, for example,

\frac{\partial r\_{12}}{\partial x\_1} = - \frac{\partial r\_{12}}{\partial x\_2}. \tag{3}

We therefore compute derivatives with respect to the atom with the lower number.

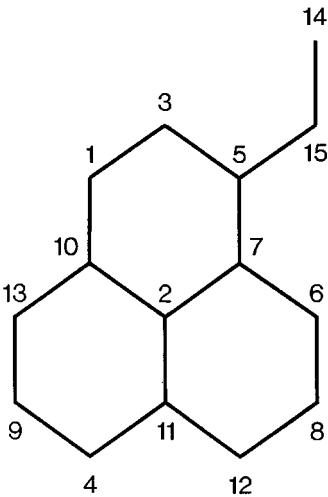


FIGURE 2. A bond network illustrating types of three- and four-body interactions.

TABLE I. Connection Table for Bond Network Shown in Figure 2.

Atom		Connections	
1	3	10	
2	7	10	11
3	1	5	
4	9	11	
5	3	7	15
6	7	8	
7	2	5	6
8	6	12	
9	4	13	
10	1	2	13
11	2	4	12
12	8	11	
13	9	10	
14	15		
15	5	14	

Table II shows two stretch interaction lists, a forward list and a reverse list. In the forward list the first atom in the interaction pair has the lower number; in the reverse list the second atom has the lower number. Also shown is a reverse index list; for example, the index for the second entry of the reverse table is 6, meaning that the second entry in the reverse table corresponds to the sixth entry of the forward list. Although only the forward list is required for treating stretch interactions, generat-

TABLE II. Forward and Reverse Stretch Interaction Lists and Reverse Index Table for Bond Network of Figure 2.

Stretch	Forward		Reverse		Index
1	1	3	3	1	1
2	1	10	5	3	6
3	2	7	7	2	3
4	2	10	7	5	9
5	2	11	7	6	11
6	3	5	8	6	12
7	4	9	9	4	7
8	4	11	10	1	2
9	5	7	10	2	4
10	5	15	11	2	5
11	6	7	11	4	8
12	6	8	12	8	13
13	8	12	12	11	16
14	9	13	13	9	14
15	10	13	13	10	15
16	11	12	15	5	10
17	14	15	15	14	17

ing the reverse list simplifies the bookkeeping for bend and other interaction types.

Storage of internal coordinates and derivatives is straightforward. These can be stored in an array, say  $B1d(i, j)$ , where  $i$  is 1, 2, or 3 for  $x$ ,  $y$ , or  $z$  and  $j$  is the stretch interaction number. A similar procedure can be followed for the six independent second derivatives.

BEND INTERACTIONS

Bend angles and their derivatives may be calculated easily from bond distances contained therein and their derivatives. For example, for a bend sequence 123, we have<sup>5</sup>

cos θ = r̂<sub>21</sub> · r̂<sub>23</sub> = (∂r<sub>12</sub>/∂x<sub>1</sub> ∂r<sub>23</sub>/∂x<sub>3</sub> + ∂r<sub>12</sub>/∂y<sub>1</sub> ∂r<sub>23</sub>/∂y<sub>3</sub> + ∂r<sub>12</sub>/∂z<sub>1</sub> ∂r<sub>23</sub>/∂z<sub>3</sub>), (4)

∂cos θ/∂q<sub>3</sub> = 1/r<sub>23</sub> (∂r<sub>23</sub>/∂q<sub>2</sub> cos θ + ∂r<sub>12</sub>/∂q<sub>1</sub>), (5)

where  $q$  is  $x$ ,  $y$ , or  $z$ . Formulas for other derivatives are similar in spirit to eq. (5).

Table III shows a list of bend interactions and, for each bend interaction, the stretches included therein. There are three general cases for treating bend interactions ( $i_1, i_2, i_3$ ) are  $i_1 < i_2 < i_3$ ,  $i_2 < i_1 < i_3$ , and  $i_1 < i_3 < i_2$ . These cases differ according to the sign in front of the bond distance derivatives. For example, the bend angle derivatives with respect to  $x$  on the third atom for the (1,3,5), (7,6,8), and (2,7,5) bend sequences may be calculated from

∂cos θ<sub>135</sub>/∂x<sub>5</sub> = 1/r<sub>35</sub> (∂r<sub>35</sub>/∂x<sub>3</sub> cos θ<sub>135</sub> + ∂r<sub>13</sub>/∂x<sub>1</sub>), (6)

∂cos θ<sub>768</sub>/∂x<sub>8</sub> = 1/r<sub>68</sub> (∂r<sub>68</sub>/∂x<sub>6</sub> cos θ<sub>768</sub> - ∂r<sub>67</sub>/∂x<sub>6</sub>), (7)

∂cos θ<sub>275</sub>/∂x<sub>5</sub> = 1/r<sub>57</sub> (-∂r<sub>57</sub>/∂x<sub>5</sub> cos θ<sub>275</sub> + ∂r<sub>27</sub>/∂x<sub>2</sub>). (8)

The bend angles themselves may be calculated from

cos θ<sub>135</sub> = - (∂r<sub>13</sub>/∂x<sub>1</sub> ∂r<sub>35</sub>/∂x<sub>3</sub> + ∂r<sub>13</sub>/∂y<sub>1</sub> ∂r<sub>35</sub>/∂y<sub>3</sub> + ∂r<sub>13</sub>/∂z<sub>1</sub> ∂r<sub>35</sub>/∂z<sub>3</sub>), (9)

TABLE III. Bend Interaction List for Bond Network of Figure 2.

Index	Atoms			Stretches		
1	1	3	5	1	6	Type I
2	3	5	7	6	9	
3	3	5	15	6	10	
4	6	8	12	12	13	
5	4	9	13	7	14	
6	1	10	13	2	15	
7	2	10	13	4	15	
8	2	11	12	5	16	
9	4	11	12	8	16	
10	3	1	10	1	2	Type II
11	7	2	10	3	4	
12	7	2	11	3	5	
13	10	2	11	4	5	
14	9	4	11	7	8	
15	7	5	15	9	10	
16	7	6	8	11	12	
17	2	7	5	3	9	Type III
18	2	7	6	3	11	
19	5	7	6	9	11	
20	1	10	2	2	4	
21	2	11	4	5	8	
22	8	12	11	13	16	
23	9	13	10	14	15	
24	5	15	14	10	17	

cos θ<sub>768</sub> = (∂r<sub>67</sub>/∂x<sub>6</sub> ∂r<sub>68</sub>/∂x<sub>6</sub> + ∂r<sub>67</sub>/∂y<sub>6</sub> ∂r<sub>68</sub>/∂y<sub>6</sub> + ∂r<sub>67</sub>/∂z<sub>6</sub> ∂r<sub>66</sub>/∂z<sub>6</sub>), (10)

cos θ<sub>275</sub> = (∂r<sub>27</sub>/∂x<sub>2</sub> ∂r<sub>57</sub>/∂x<sub>5</sub> + ∂r<sub>27</sub>/∂y<sub>2</sub> ∂r<sub>57</sub>/∂y<sub>5</sub> + ∂r<sub>27</sub>/∂z<sub>2</sub> ∂r<sub>57</sub>/∂z<sub>5</sub>), (11)

and so the bend angles may be calculated in two loops, one for type I and one for types II and III.

A complete list of bend interactions, including the stretch interactions contained therein, for the bond network of Figure 2 is shown in Table III. Types I, II, and III correspond to the use of eqs. (6), (7), and (8). Type I interactions are generated from the forward and reverse stretch interaction lists, type II from the reverse list, and type III from the forward list. For example, the first type I bend is obtained by noting that atom 3 is the first atom of the sixth entry of the forward stretch interaction

list and the first entry in the reverse list. The first type II bend is obtained from the first two entries in the forward interaction list, for which the first atom is atom 1. The first three type III bends are obtained from the third through fifth entries in the reverse stretch interaction list, for which the first atom in the interaction pair is atom 7.

If the bend interaction table is separated into regions of types I, II, and III, then which signs to place in front of derivatives of bond distances is automatically accounted for. The bend interaction table can be generated once and used throughout the rest of the calculation. This makes it unnecessary to compare atom numbers within the loops in which bend angles and their derivatives are calculated, making for a more efficient calculation. This method can be used with any type of bond network and the same code used in simulations of a polymer chain can be used without modification in, for example, carbon nanotube simulations.

The entries within each type in the bend interaction table are arranged in order of increasing cen-

ter atom number. For example, the number of the center atom for the first type I bend interaction is 3, the next two are 5, and so on. This ordering will become important in the discussion of torsion interactions. The process for generating bend interaction tables is shown in Figure 3.

Some common force fields, such as MM2,<sup>8</sup> are parameterized by  $\theta$  rather than  $\cos \theta$ . Derivatives of  $\theta$  can be calculated directly from derivatives of  $\cos \theta$ :

$$\frac{\partial \theta}{\partial q_i} = -\frac{1}{\sin \theta} \frac{\partial \cos \theta}{\partial q_i} \quad (12)$$

$$\frac{\partial^2 \theta}{\partial q_i \partial q_j} = -\frac{1}{\sin \theta} \left( \frac{\partial^2 \cos \theta}{\partial q_i \partial q_j} + \cos \theta \frac{\partial \theta}{\partial q_i} \frac{\partial \theta}{\partial q_j} \right). \quad (13)$$

Derivatives of  $\theta$  can be computed from eqs. (12) and (13) in a single loop, without the need for categorizing into types.

#### Stretch interactions:

Generate forward and reverse interaction lists.

Generate reverse index table.

#### Bend interactions:

Type I -- For each atom b,

{a<sub>i</sub>} atoms connected to b from reverse stretch interaction list.

{c<sub>j</sub>} atoms connected to b from forward stretch interaction list.

Type I bends {(a<sub>i</sub>, b, c<sub>j</sub>)}.

Type II -- For each atom b,

{c<sub>j</sub>} from forward stretch interaction list.

Type II bends {(c<sub>i</sub>, b, c<sub>k</sub>), i < k}.

Type III -- For each atom b,

{a<sub>j</sub>} from reverse stretch interaction list.

Type III bends {(a<sub>j</sub>, b, a<sub>k</sub>), j < k}.

#### Torsion interactions:

Type Ia -- For each type III bend (a, b, c),

Find type II bends with center atom c.

Find type II bends {(b, c, d<sub>i</sub>)}.

Type Ia torsions {(a, b, c, d<sub>i</sub>)}.

Type Ib -- For each type I bend (a, b, c),

Find type I bends with center atom c.

Find type I bends {(b, c, d<sub>i</sub>)}.

Type Ib torsions {(a, b, c, d<sub>i</sub>)}.

Other torsions found in a similar manner.

**FIGURE 3.** Process for generation of stretch, bend, and torsion (types Ia and Ib) interaction lists.

The indexing scheme for  $\cos \theta$  and its derivatives is similar to that for the stretch interactions. Let  $C(j)$  be the array of  $\cos \theta$ , where  $j$  is the bend interaction number as shown in Table III. The very first bend contains stretch interactions 1 and 6. Therefore, by analogy to eq. (9) we have

$$C(1) = -[B1d(1,1)B1d(1,6) + B1d(2,1)B1d(2,6) + B1d(3,1)B1d(3,6)].$$

Other  $\cos \theta$  may be calculated in a similar manner. There are six independent first derivatives. Let  $kmn$  be the atom numbers for a bend interaction. Then the first derivatives may be stored in an array, say  $C1d(i, j)$ , where  $i = 1, 2, \dots, 6$  denotes derivatives with respect to  $x_k, y_k, z_k, x_n, y_n, z_n$ , and  $j$  denotes the interaction number. The 21 independent second derivatives may be stored in a similar manner. However, it should be noted that for  $\theta$ , but not for  $\cos \theta$ , there is an additional relation

$$\frac{\partial^2 \theta}{\partial q_1 \partial q'_3} = \frac{\partial^2 \theta}{\partial q'_1 \partial q_3}, \tag{14}$$

where  $q$  and  $q'$  are  $x, y$ , or  $z$ , which reduces the number of independent second derivatives from 21 to 18.

TORSIONAL INTERACTIONS

The formulas for torsion angles and their derivatives can be written in terms of derivatives of bond distances and bend angles. Thus, it is necessary to account not only for the orders of the atom numbers in stretch interactions, but also whether the atom numbers in the bend interactions appear in the same or in reverse order as in the bend interaction tables. To make this clearer, we first list formulas for the torsion angle of a four-atom sequence labeled 1234 and several of its derivatives<sup>5</sup>:

$$\begin{aligned} \cos \tau &= \frac{1}{\sin \theta_{123} \sin \theta_{234}} \left( \cos \theta_{123} \cos \theta_{234} - \frac{\partial r_{12}}{\partial x_2} \frac{\partial r_{34}}{\partial x_4} \right. \\ &\quad \left. - \frac{\partial r_{12}}{\partial y_2} \frac{\partial r_{34}}{\partial y_4} - \frac{\partial r_{12}}{\partial z_2} \frac{\partial r_{34}}{\partial z_4} \right), \tag{15} \end{aligned}$$

$$\frac{\partial \cos \tau}{\partial q_1} = a_{123} \left( \cos \tau \frac{\partial \theta_{123}}{\partial q_3} - \frac{\partial \theta_{234}}{\partial q_2} \right) \tag{16}$$

$$\begin{aligned} \frac{\partial^2 \cos \tau}{\partial q_2 \partial q'_2} &= c_{123} \frac{\partial^2 \cos \tau}{\partial q_1 \partial q'_2} - b_{432} \frac{\partial^2 \cos \tau}{\partial q_4 \partial q'_2} \\ &\quad - \frac{1}{r_{23}} \left\{ \frac{\partial \cos \tau}{\partial q_2} \frac{\partial r_{23}}{\partial q'_2} \right. \\ &\quad + r_{12} \frac{\partial \cos \theta_{123}}{\partial q'_3} \frac{\partial \cos \tau}{\partial q_1} \\ &\quad \left. + r_{34} \frac{\partial \cos \theta_{234}}{\partial q'_2} \frac{\partial \cos \tau}{\partial q_4} \right\} \tag{17} \end{aligned}$$

where  $a_{123}$ ,  $c_{123}$ , and  $b_{432}$  are constants and  $q$  and  $q'$  are independent and may be  $x, y$ , or  $z$ .

To determine the possible varieties of the implementation of torsion formulas, we first consider the permutations of four atom numbers 1, 2, 3, and 4, as shown in Table IV. Only 12 of the 24 permutations are listed in order to avoid overcounting (for example, 4321 is equivalent to 1234). The next major column lists stretch and bend types. For example, the stretch types for the fifth permutation, 1342, are listed as  $++-$ , meaning that the first two bonds, 13 and 34, are in forward order according to atom number and the third bond, 42, in reverse order. The first bend, 134, is of type I and is in forward order. The second bend, 342, is of type II and is in reverse order (i.e., the bend interaction table would contain 243 not 342).

We are now ready to segregate the torsion computations into separate cases. Because the formula for torsion angle involves bond distance derivatives, eq. (15) can be split into two cases much as eq. (5) was split into three. To calculate first

TABLE IV. Four Atom Permutations and Their Stretch, Bend, and Torsion ( $\cos \tau$ ) Types.

Permutation	Stretch, Bend	Torsion	Remarks
1 3 2 4	$+-+, 3^+2^+$	Ia	Acts same in MD, not MM.
1 2 3 4	$+++, 1^+1^+$	Ib	
1 2 4 3	$++-, 1^+3^+$	IIa	Acts same in MD and MM.
2 1 3 4	$-++, 2^+1^+$	IIb	
1 3 4 2	$++-, 1^+3^-$	III	Acts same in MD and MM.
2 3 4 1	$++-, 1^+3^-$		
1 4 2 3	$+-+, 3^+2^-$	IV	Acts same in MD and MM.
2 4 1 3	$+-+, 3^-2^-$	V	
3 4 1 2	$+-+, 3^-2^-$		
2 3 1 4	$+-+, 3^-2^+$	VI	Acts same in MD and MM.
3 1 2 4	$-++, 2^-1^+$	VII	
4 1 2 3	$-++, 2^-1^+$		

derivatives of torsion angles requires first derivatives of bend angles with respect to end atom coordinates, as shown in eq. (16). Thus, the first derivative formulas can be split into four cases corresponding to the orientations (forward or reverse) of the bend angles. Equation (17) shows that second derivatives of torsion angles require accounting for not only the orientations of the bend angles, but also the sign in front of derivatives of the middle bond distance.

The permutations in Table IV are segregated according to torsion type. Types Ia and Ib have the same bend angle orientations but different orientations for the middle bond; thus, they act the same in molecular dynamics (MD) simulations but not in molecular mechanics (MM) or normal coordinate analysis (NCA) calculations, where second derivatives are required. Types IIa and IIb have the same calculation sequence in all molecular simulations involving up to second derivatives; they differ only in the types of bends used to generate them.

The permutations in Table IV are arranged in such an order as to yield the longest possible loops when calculating first and second derivatives. For example, when calculating first derivatives, types Ia through IIb can be performed in the same loop (because every bend angle orientation is forward), as can types IV and V and also VI and VII. When calculating second derivatives, types Ib through IIb can be performed in the same loop.

Table V shows a torsion interaction list for the bond network of Figure 2. The numbers of the atoms in the bond network were assigned to ensure that torsions of all types were present. The process for generating the first two types of torsion interactions is shown in Figure 3; the rest of the table is generated in a similar manner. For example, the two bends within a type Ia torsion are of types III and II, respectively, both in forward order. The first type III bend is (2, 7, 5). The second bend is of type II and has atom number 5 in the center. Thus, the second to last type II bend, (7, 5, 15), completes the torsion sequence (2, 7, 5, 15). Other entries in the torsion interaction list follow in a similar manner.

The above treatment is for torsional potentials parameterized by  $\cos \tau$ . However, it is best not to calculate derivatives of  $\tau$  from derivatives of  $\cos \tau$  because of the well-known  $1/\sin \tau$  singularity.<sup>6</sup> It is possible to calculate these derivatives directly. From the second derivative formulas we see that torsional interactions parameterized by  $\tau$  may be categorized in the same manner as those parame-

terized by  $\cos \tau$ . However, there is one important difference in the treatment of first derivatives. For a torsion sequence 1234, the end atom first derivatives are<sup>5,6</sup>

$$\frac{\partial \tau}{\partial q_1} = a_{123} \left( \frac{\partial r_{12}}{\partial s_1} \frac{\partial r_{23}}{\partial u_3} - \frac{\partial r_{12}}{\partial u_1} \frac{\partial r_{23}}{\partial s_3} \right), \quad (18)$$

$$\frac{\partial \tau}{\partial q_4} = -a_{432} \left( \frac{\partial r_{23}}{\partial s_3} \frac{\partial r_{34}}{\partial u_3} - \frac{\partial r_{23}}{\partial u_3} \frac{\partial r_{34}}{\partial s_3} \right), \quad (19)$$

where  $a_{123}$  and  $a_{432}$  are constants. Middle atom derivatives can be calculated from the end atom derivatives. It should be noted that atom 1 derivatives depend on quantities within the bend angle 123 and are independent of atom 4; a similar statement holds for atom 4. Therefore, a complete table of end atom first derivatives of  $\tau$  may be stored in a table with a size on the order of the number of bend interactions, not the number of torsion interactions. This can economize force calculations considerably in complex bond networks. Of course, for each torsion interaction  $\tau$  and  $\partial V / \partial \tau$  must still be calculated.

The indexing and lookup procedure for  $\cos \tau$  and its derivatives can be made similar to those shown above for stretch and bend interactions. However, the number of independent first and second derivatives are nine and 45, respectively. One additional level of complexity to account for is the order of the atom numbers (for example, 123 versus 321) in the bend interactions that are contained within the torsion interaction. Similar considerations hold for  $\tau$ . However, because

$$\frac{\partial^2 \tau}{\partial q_1 \partial q'_4} = 0, \quad (20)$$

the number of independent second derivatives to be calculated is 36 not 45.

It is almost never necessary to store torsion angle derivatives because they do not appear as intermediates except in special cases such as if cross terms involving five or more atoms (such as torsion-improper torsion-torsion) are present. In this case, if care is taken to store only those derivatives that appear in the cross terms, the memory requirements can be minimized.

### MM2, MM3, AND MM4 OUT OF PLANE BEND INTERACTIONS

In the MM2<sup>8</sup> and MM3<sup>9,10</sup> force fields, deviation from an  $sp^2$  equilibrium planar geometry is described by the movement of the central atom

TABLE V.  
Torsion Interaction List for Bond Network of Figure 2.

No.		Atoms			Stretches			Bends		
1	2	7	5	15	3	9	10	17	15	Type Ia
2	2	7	6	8	3	11	12	18	16	
3	5	7	6	8	9	11	12	19	16	
4	1	10	2	11	2	4	5	20	13	
5	1	3	5	7	1	6	9	1	2	Type Ib
6	1	3	5	15	1	6	10	1	3	
7	3	5	7	6	5	9	11	2	19	Type IIa
8	3	5	15	14	6	10	17	3	24	
9	6	8	12	11	12	13	16	4	22	
10	4	9	13	10	7	14	15	5	23	
11	3	1	10	13	1	2	15	10	6	Type IIb
12	7	2	10	13	3	4	15	11	7	
13	7	2	11	12	3	5	16	12	8	
14	10	2	11	12	4	5	16	13	8	
15	9	4	11	12	7	8	16	14	9	
16	7	6	8	12	11	12	13	16	4	
17	3	5	7	2	6	9	3	2	17	Type III
18	1	10	13	9	2	15	14	6	23	
19	2	10	13	9	4	15	14	7	23	
20	2	11	12	8	5	16	13	8	22	
21	4	11	12	8	8	16	13	9	22	
22	1	10	2	7	2	4	3	20	11	Type IV
23	2	11	4	9	5	8	7	21	14	
24	2	10	1	3	4	2	1	20	10	Type V
25	4	11	2	7	8	5	3	21	12	
26	4	11	2	10	8	5	4	21	13	
27	14	15	5	7	17	10	9	24	15	
28	5	7	2	10	9	3	4	17	11	Type VI
29	5	7	2	11	9	3	5	17	12	
30	6	7	2	10	11	3	4	18	11	
31	6	7	2	11	11	3	5	18	12	
32	6	7	5	15	11	9	10	19	15	
33	10	1	3	5	2	1	6	10	1	Type VII
34	11	2	10	13	5	4	15	13	7	
35	11	4	9	13	8	7	14	14	5	

(atom 4 in Fig. 1d). Out of plane bend energies are parameterized by three projected in-plane angles of the type  $\theta_{k4'l'}$ , and three out of plane bend angles of the type  $\theta_{4k4'}$ , where 4' is the projection of point 4 into the 123 plane. In the MM4<sup>11,12</sup> force field, three interlocking wag (also called improper dihedral) angles of the sort shown in Figure 1c are used instead.

The formulas for all of these internal coordinates and their derivatives involve stretch and bend derivatives and are shown elsewhere.<sup>13</sup> However, because in each case every possible out

of plane angle and its derivatives are calculated, the computation can be segregated into four types, depending on whether the center atom has the largest, second largest, third largest, or smallest number. For example, if the center atom has the largest number, then every bend interaction is of type III.

A set of improper torsion or out of plane bend interactions can be generated from the bend interaction tables for each atom that is the center atom in three bend interactions. For example, atom 7 is the center atom in the first three type III bends.



**TABLE VI.**  
**Improper Torsion / Out of Plane Bend Interaction List for Bond Network of Figure 2.**

No.	Atoms (center first)					Stretches		Bends			
1	2	7	10	11	3	4	5	11	12	13	Type I
2	5	3	7	15	6	9	10	2	3	15	Type II
3	10	1	2	13	2	4	15	6	7	20	Type III
4	11	2	4	12	5	8	16	8	9	21	
5	7	2	5	6	3	9	11	17	18	19	Type IV

Placing the correct bend and stretch numbers in the improper torsion interaction table is straightforward. The full interaction table is listed in Table VI.

In the MM2 and MM3 force field, a single  $sp^2$  hybridized atom and its three satellite atoms has six in-plane and out of plane bend angles. For each of these there are nine independent first derivatives and 45 independent second derivatives, making a total of 54 first derivatives and 270 second derivatives. In the MM4 force field, there the three improper dihedral angles; each has nine independent first derivatives and 45 independent second derivatives, giving rise to a total of 27 first derivatives and 135 second derivatives. To reduce memory requirements, it is best to use each set of derivatives once (to treat the out of plane interactions and any cross terms involving them) and not to store them for further use.

## Results and Discussion

The general bond network method has been used successfully in several simulations involving polymer chains or carbon nanotubes. Most recently it was used in calculations involving the newly developed internal coordinate quantum Monte Carlo<sup>14,15</sup> (ICQMC) method, which requires force calculations resembling those in MD simulations. One useful test of the internal coordinate derivative portion of the ICQMC code was to plug it into an MD force subroutine and verify that the resulting MD code conserved energy.

The total storage space required for internal coordinates and their derivatives depends on the connectivity of the bond network. For example, a linear polymer chain would have fewer bend and torsion interactions than a carbon nanotube with the same number of atoms, which in turn would have fewer interactions than a diamondoid bond

network. Storage requirements also depend on the functional form of the potential energy surface. For example, if the potential energy surface contains only bend, stretch, and torsion interactions with no cross terms, then only bend and stretch derivatives should be stored because derivatives of torsion angles are used only once. In general, we have found that a very highly connected bond network with  $N$  atoms will have up to  $2N$  stretch interactions,  $4N$  bend interactions at most. It is usually not necessary to store derivatives of four-body internal coordinates because they do not serve as intermediates for further calculations except in the special case of cross terms. In this case, if care is taken to store only those torsion angle derivatives used in cross terms, the memory requirements can be minimized. The number of interactions, plus the number of independent derivatives per interaction, can be used to estimate storage requirements. For systems with very large numbers of atoms, it is possible to divide the effort in order to not overflow the memory. For example, interactions containing atoms 1–1000 can be treated first, then those containing atoms 1001–2000 but not the first 1000, and so on.

The treatment presented here so far has not considered the presence of atoms of different chemical types. It is best in such cases for each atom type to take a single block in the connection table; this often simplifies the calculations further. For example, in a fully atomistic polyethylene chain, if the carbon atoms are numbered below the hydrogen atoms, there are only two varieties of CCH interaction (types I and II). After following the procedure presented here, the table of type I bends would include CCC interspersed with CCH bends. By reordering the type I bend table to contain first CCC bends then CCH bends, calculation of potential energies or forces could be optimized further. Similar procedures could be followed for type II bends.

This strategy may be modified in special cases. In simulations of fully atomistic polyethylene crystals, it is best to order the connection table first by chain number and then by atom type. Such a numbering scheme makes it easier to parallelize the simulation code for large crystals because spatial domains are naturally distributed by entire chains. In addition, if each chain is of the same length, interaction lists need to be generated only for one chain.

## Conclusions

By starting from a connection table, it is possible to efficiently generate lists of multibody interactions and, for each interaction, a list of internal coordinates contained therein. The method presented here works for any bond network, no matter how complex. Using this method makes it possible to take full advantage of formulas for internal coordinates and their derivatives that are written in terms of simpler internal coordinates and their derivatives. In addition, it greatly speeds the adaptation of molecular simulation codes for modifications of a potential energy surface or bond network type.

This article described a method for treating general bond networks that essentially standardizes the portion of code devoted to the calculation of bonded internal coordinates and their derivatives and that ensures the most optimal calculation sequence. The treatment of bend, torsion, wag, and out of plane coordinates were described in detail. The general bond network method should work well for other internal coordinates not considered here.

Code for calculating internal coordinates and their derivatives can be used with little or no modification for different molecular simulation techniques (MD, MM, ICQMC, etc.) and for different bond network types such as polymer chains, carbon nanotubes, and diamondoid structures. The general bond network method will undoubtedly

serve well for other molecular simulation techniques as they are developed.

## Acknowledgments

This research was sponsored by the Division of Materials Sciences, Office of Basic Energy Sciences, U. S. Department of Energy, under Contract DE-AC05-96OR22464 with Lockheed Martin Energy Research Corp. It was also supported in part by an appointment to the Oak Ridge Postdoctoral Research Associates Program administered jointly by the Oak Ridge National Laboratory and the Oak Ridge Institute for Science and Education.

## References

1. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford, 1987.
2. K. J. Miller, R. J. Hinde, and J. Anderson, *J. Comput. Chem.*, **10**, 63 (1989).
3. K. J. Miller, *J. Comput. Chem.*, **11**, 336 (1990) and references therein.
4. D. W. Noid, B. G. Sumpter, B. Wunderlich, and G. A. Pfeffer, *J. Comput. Chem.*, **11**, 236 (1990).
5. R. E. Tuzun, D. W. Noid, and B. G. Sumpter, *Macromol. Theory Simul.*, **6**, 771 (1996) and references therein.
6. H. Bekker, H. J. C. Berendsen, and W. F. vanGunsteren, *J. Comput. Chem.*, **16**, 527 (1995) and references therein.
7. B. Jung, *Makromol. Chem., Theory Simul.*, **2**, 673 (1993).
8. N. L. Allinger and A. Pathiaseril, *J. Comput. Chem.*, **8**, 1225 (1987).
9. N. L. Allinger, F. Li, and L. Yan, *J. Comput. Chem.*, **11**, 848 (1990).
10. N. L. Allinger, F. Li, L. Yan, and J. C. Tai, *J. Comput. Chem.*, **11**, 868 (1990).
11. N. Nevins, K. Chen, and N. L. Allinger, *J. Comput. Chem.*, **17**, 669 (1996).
12. N. Nevins and N. L. Allinger, *J. Comput. Chem.*, **17**, 695 (1996).
13. R. E. Tuzun, D. W. Noid, and B. G. Sumpter, *J. Comput. Chem.*, to appear.
14. R. E. Tuzun, D. W. Noid, and B. G. Sumpter, *J. Chem. Phys.*, **105**, 5494 (1996).
15. R. E. Tuzun, D. W. Noid, and B. G. Sumpter, *Nanotechnology*, to appear.